Homework 4
APPM/MATH 4650 Fall '20 Numerical Analysis

Due date: Monday, October 5, before midnight, via Gradescope. **Instructor**: Prof. Becker
Theme: Splines

Instructions Collaboration with your fellow students is OK and in fact recommended, although direct copying is not allowed. The internet is allowed for basic tasks (e.g., looking up definitions on wikipedia) but it is not permissible to search for proofs or to *post* requests for help on forums such as `http://math.stackexchange.com/` or to look at solution manuals. Please write down the names of the students that you worked with.

An arbitrary subset of these questions will be graded.

Turn in a PDF (either scanned handwritten work, or typed, or a combination of both) to **Gradescope**, using the link to Gradescope from our Canvas page. Gradescope recommends a few apps for scanning from your phone; see the Gradescope HW submission guide.

We will primarily grade your written work, and
source code is *not* necessary except for when we *explicitly* ask for it (and you can use any language you want). If not specifically requested as part of a problem, you may include it at the end of your homework if you wish (sometimes the graders might look at it, but not always; it will be a bit easier to give partial credit if you include your code).

Problem 1: Is the barycentric formula ill-conditioned?

a) Consider $f(x) = \frac{1}{1-x}$.
   i. Find the relative condition number of evaluating f. Is this good or bad when $x \approx 1$?
   ii. Actually evaluate f for $x = 1 - 10^{-13}$ on the computer, using any software/language that uses floating point numbers[1], and report your answer and the error.
   iii. About how many correct digits do you have? Is this expected?
   iv. If you evaluate f for $x = 1 - 2^{-43} \approx 1 - 1.13 \cdot 10^{-13}$, how many correct digits do you have now? (Don't forget your true answer has changed)

b) The barycentric formula for Lagrange interpolation looks something like

$$p(x) = \sum_{i=0}^{n} \frac{\frac{w_i}{x-x_i} f_i}{\frac{w_i}{x-x_i}}$$

which might make you nervous when $x \approx x_i$ for some i. Let's analyze a simplified version of this formula that still retains all the interesting (and worrisome?) behavior:

$$f(x) = \frac{\frac{c}{1-x} + d}{\frac{1}{1-x} + 1}$$

for some constant d and $c \neq 0$.

Find the relative condition number κ_f for $x = 1$.

[1] Do *do not* use a symbolic computation framework like Mathematica unless you explicitly convert to floating point numbers by, e.g., writing $x = 1. - 10.^{-13.}$ where the decimal dots tell it to use floating point. Tools like graphing calculators and Wolfram Alpha are also probably to be avoided since they try to guess what you want, and could be doing it in either exact arithmetic or floating point.

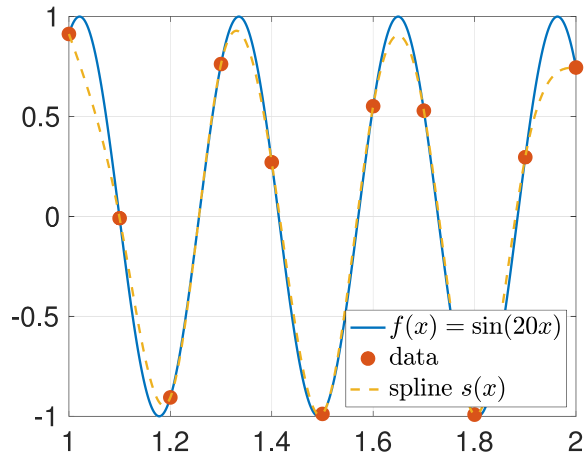

Figure 1: Example of spline for Problem 2

Problem 2: **Splines** For this problem, we will do cubic splines. You do not need to implement these yourselves; in Matlab, you can use the curve-fitting toolbox (should be free with our campus subscription; you can also remotely use CU desktops if you need to) and the functions `csape` (to find the spline) and `fnval` (to evaluate the spline); in Python, use `scipy.interpolate.CubicSpline`.

We'll use the function

$$f(x) = \sin(20x)$$

and the interval $[1, 2]$. This function will give us the values to use on the nodes. We will only consider equispaced nodes $[x_0 = 1, x_1 = 1+h, \ldots, x_n = 1+nh = 2]$ for $h = 1/n$, and will choose different values for n.

a) Create both the **natural** cubic spline as well as the **not-a-knot** cubic spline (using the library code; you do NOT need to program the details yourself). Denote the spline by $s(x)$. We want to measure the error $|s(x) - f(x)|$ for generic points x. For this sub-problem, sample 10^5 points uniformly at random from the interval $[1.01, 1.99]$ (use Matlab's `rand` or Python's `numpy.random.rand`), and report the maximum error $|s(x) - f(x)|$ for these x values. Plot this error, for both types of cubic splines, on a figure as a function of how many points n (or $n+1$) are used for nodes. Use the plot to graphically demonstrate the order of convergence of the error. *Hint*: Make sure you have a wide enough range of n values to make this convincing and make wise choices about whether axes should be linear or logarithmic.

b) Is the order of convergence you found above expected? Discuss briefly

c) Repeat problem (a) but this time sample the 10^5 test points uniformly from $[1, 2]$, and show the same kind of plot as in (a). Are the results the same or different? Discuss briefly.

d) Repeat problem (a), but get the values from the function g, where

$$g(x) = \begin{cases} f(x) & x < 1.3 \\ f(2.6 - x) & x \geq 1.3 \end{cases}.$$

Show the same kind of plot as before. Are the results the same or different? Discuss briefly.